title: “Analysis of U.S. Storm Event Data and the Impact on Population Health and the Economy” author: “Erik Donathan” date: “30 January, 2023” —

Environment Setup

Load packages used in this analysis.

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

```
## Loading required package: ggplot2
```

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

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

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

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

Display session information.

```
sessionInfo()
```

```
## R version 4.2.2 (2022-10-31)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Ventura 13.1
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRblas.0.d
ylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRlapack.d
ylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] xtable_1.8-4  dplyr_1.0.10  ggplot2_3.4.0
##
## loaded via a namespace (and not attached):
##  [1] bslib_0.4.2      compiler_4.2.2   pillar_1.8.1     jquerylib_0.1.4
##  [5] tools_4.2.2      digest_0.6.31    jsonlite_1.8.4   evaluate_0.20
##  [9] lifecycle_1.0.3  tibble_3.1.8     gtable_0.3.1     pkgconfig_2.0.3
## [13] rlang_1.0.6      cli_3.6.0        DBI_1.1.3        rstudioapi_0.14
## [17] xfun_0.36        fastmap_1.1.0    withr_2.5.0      knitr_1.42
## [21] generics_0.1.3   vctrs_0.5.2      sass_0.4.5       grid_4.2.2
## [25] tidyselect_1.2.0 glue_1.6.2       R6_2.5.1         fansi_1.0.4
## [29] rmarkdown_2.20   magrittr_2.0.3   scales_1.2.1     htmltools_0.5.4
## [33] assertthat_0.2.1 colorspace_2.1-0 utf8_1.2.2       munsell_0.5.0
## [37] cachem_1.0.6
```

Load Data

```
setwd("~/Desktop")
stormDataFileURL <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.cs
v.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

```
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"
```

```
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 ...
```

```
head(stormData)
```

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

Data Processing

Create Subset of Data

```
stormDataTidy <- subset(stormData, EVTYPE != "?"
                                   &
                                   (FATALITIES > 0 | INJURIES > 0 | PROPDMG > 0 | CROPDM
G > 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
```

Clean Event Type Data

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

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

```
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$EVTYPE <- gsub('.*RECORD HIGH TEMPERATURES.*', 'HEAT', stormDataTidy$EVTYP
E)
# HYPOTHERMIA/EXPOSURE
stormDataTidy$EVTYPE <- gsub('.*HYPOTHERMIA.*', 'HYPOTHERMIA/EXPOSURE', stormDataTidy$EV
TYPE)
# LANDSLIDE
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)
```

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

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

Clean Date Data

```
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

```
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
```

```
# 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$PROP_COST <- with(stormDataTidy, as.numeric(PROPDMG) * sapply(PROPDMGEXP,
getMultiplier))/10^9
stormDataTidy$CROP_COST <- with(stormDataTidy, as.numeric(CROPDMG) * sapply(CROPDMGEXP,
getMultiplier))/10^9
```

Summarize Data

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

```
healthImpactData <- aggregate(x = list(HEALTH_IMPACT = stormDataTidy$FATALITIES + stormD
ataTidy$INJURIES),
                                  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.

```
damageCostImpactData <- aggregate(x = list(DAMAGE_IMPACT = stormDataTidy$PROP_COST + sto
rmDataTidy$CROP_COST),
                                  by = list(EVENT_TYPE = stormDataTidy$EVTYPE),
                                  FUN = sum,
                                  na.rm = TRUE)
damageCostImpactData <- damageCostImpactData[order(damageCostImpactData$DAMAGE_IMPACT, d
ecreasing = TRUE),]
```

Results

Event Types Most Harmful to Population Health

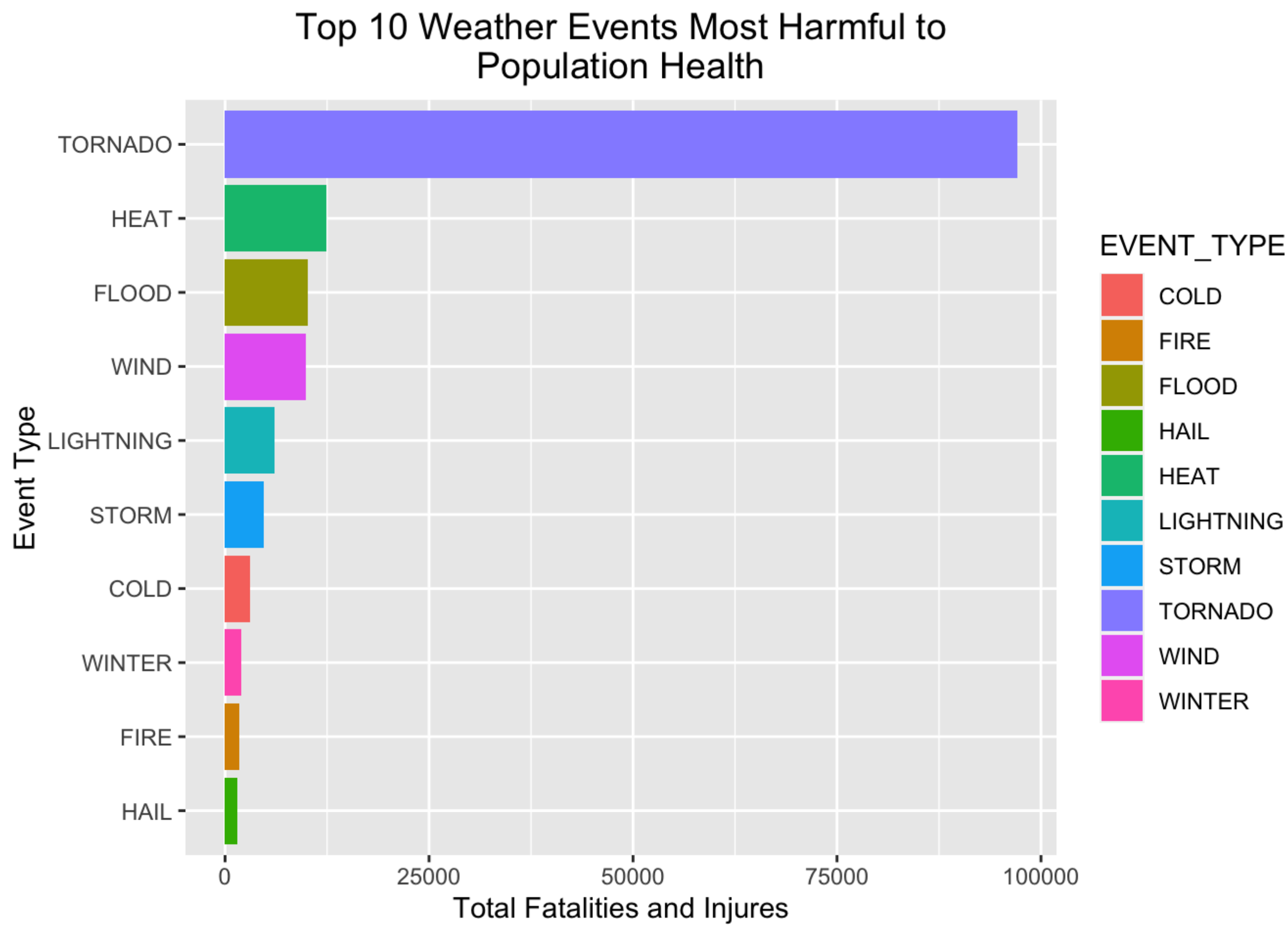
Fatalities and injuries have the most harmful impact on population health. The results below display the 10 most harmful weather events in terms of population health in the U.S.

```
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%"')
```

Top 10 Weather Events Most Harmful to Population Health

EVENT_TYPE	HEALTH_IMPACT
TORNADO	97075.00
HEAT	12392.00
FLOOD	10127.00
WIND	9893.00
LIGHTNING	6049.00
STORM	4780.00
COLD	3100.00
WINTER	1924.00
FIRE	1698.00
HAIL	1512.00

```
healthImpactChart <- ggplot(head(healthImpactData, 10),
                            aes(x = reorder(EVENT_TYPE, HEALTH_IMPACT), y = HEALTH_IMPAC
T, fill = EVENT_TYPE)) +
                            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 H
ealth")
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.

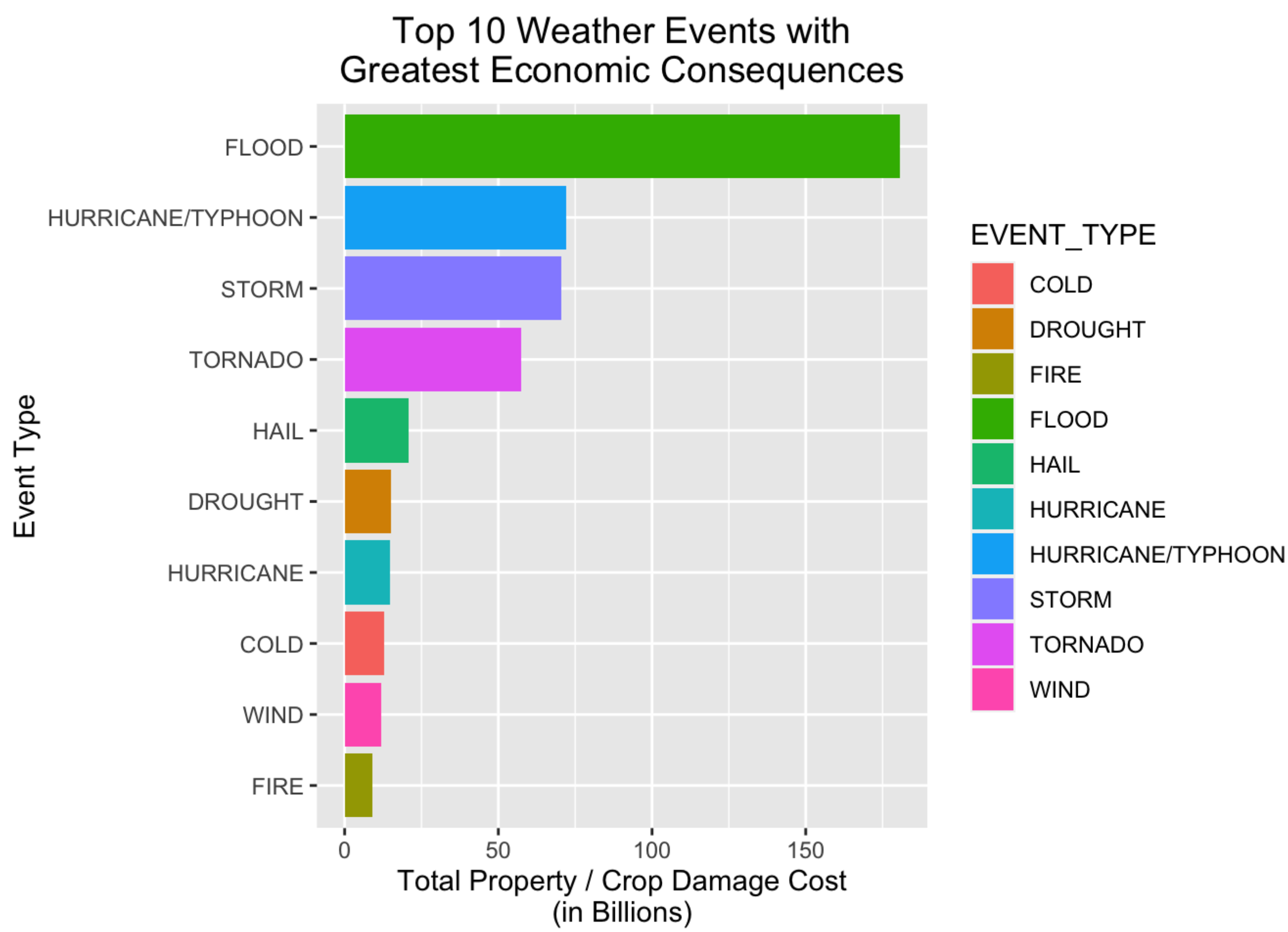
```
print(xtable(head(damageCostImpactData, 10),
             caption = "Top 10 Weather Events with Greatest Economic Consequences"),
             caption.placement = 'top',
             type = "html",
             include.rownames = FALSE,
             html.table.attributes='class="table-bordered",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

Top 10 Weather Events with Greatest Economic Consequences

EVENT_TYPE	DAMAGE_IMPACT
FLOOD	180.58
HURRICANE/TYPHOON	71.91
STORM	70.45
TORNADO	57.43
HAIL	20.74
DROUGHT	15.02
HURRICANE	14.61
COLD	12.70
WIND	12.01

FIRE	8.90

```
damageCostImpactChart <- ggplot(head(damageCostImpactData, 10),
                            aes(x = reorder(EVENT_TYPE, DAMAGE_IMPACT), y = DAMAGE_IMPAC
T, fill = EVENT_TYPE)) +
                            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 Conse
quences")
print(damageCostImpactChart)
```



Conclusion

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

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