

Environment Setup

Load Data

Data Processing

Results

Conclusion

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

Kirtan Somani

Course Project

Reproducible Research Course Project 2

Peer-graded Assignment

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

## Registered S3 methods overwritten by 'ggplot2':
##   method      from
##   [.quosures   rlang
##   c.quosures   rlang ##   print.quosures rlang

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 3.6.0 (2019-04-26)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Sierra 10.12.6 ##
## Matrix products: default ## BLAS:
/Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.
0.dylib ## LAPACK:
/Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.k.dylib
## ## 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_0.8.1 ggplot2_3.1.1
## ## loaded via a namespace (and not
attached):
## [1] Rcpp_1.0.1      knitr_1.23      magrittr_1.5    tidyselect_0.2.5
## [5] munsell_0.5.0   colorspace_1.4-1 R6_2.4.0        rlang_0.3.4
## [9] stringr_1.4.0   plyr_1.8.4      tools_3.6.0     grid_3.6.0
## [13] packrat_0.5.0   gtable_0.3.0    xfun_0.7        withr_2.1.2
## [17] htmltools_0.3.6 assertthat_0.2.1 yaml_2.2.0       lazyeval_0.2.2
## [21] digest_0.6.18   tibble_2.1.1    crayon_1.3.4    purrr_0.3.2
## [25] glue_1.3.1      evaluate_0.13   rmarkdown_1.12  stringi_1.4.3
## [29] compiler_3.6.0  pillar_1.4.0    scales_1.0.0    pkgconfig_2.0.2
```

Load Data

```
setwd("~/repos/coursera/data-science-specialization-githubassignments/reproducible-research-course-project-2")
stormDataURL <- "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 = stormDataURL, 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"         "EVTTYPE"       "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 : Factor w/ 16335 levels "1/1/1966 0:00:00",...: 6523 6523 4242 1
1116 2224 2224 2260 383 3980 3980 ...
## $ BGN_TIME : Factor w/ 3608 levels "00:00:00 AM",...: 272 287 2705 1683 2584
3186 242 1683 3186 3186 ...
## $ TIME_ZONE : Factor w/ 22 levels "ADT","AKS","AST",...: 7 7 7 7 7 7 7 7 7 7
...
## $ COUNTY : num 97 3 57 89 43 77 9 123 125 57 ...
## $ COUNTYNAME: Factor w/ 29601 levels "", "5NM E OF MACKINAC BRIDGE TO PRESQUE
ISLE LT MI",...: 13513 1873 4598 10592 4372 10094 1973 23873 24418 4598 ...
## $ STATE : Factor w/ 72 levels "AK","AL","AM",...: 2 2 2 2 2 2 2 2 2 2 ...
## $ EVTTYPE : Factor w/ 985 levels " HIGH SURF ADVISORY",...: 834 834 834 8
34 834 834 834 834 834 834 ...
## $ BGN_RANGE : num 0 0 0 0 0 0 0 0 0 0 ...
## $ BGN_AZI : Factor w/ 35 levels "", " N"," NW",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ BGN_LOCATI: Factor w/ 54429 levels "", " Christiansburg",...: 1 1 1 1 1 1 1
1 1 1 ...
## $ END_DATE : Factor w/ 6663 levels "", "1/1/1993 0:00:00",...: 1 1 1 1 1 1 1
1 1 1 ...
## $ END_TIME : Factor w/ 3647 levels "", " 0900CST",...: 1 1 1 1 1 1 1 1 1 1
...
## $ 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 : Factor w/ 24 levels "", "E","ENE","ESE",...: 1 1 1 1 1 1 1 1 1 1
...
## $ END_LOCATI: Factor w/ 34506 levels "", " CANTON"," TULIA",...: 1 1 1 1 1 1 1
1 1 1 ...
## $ 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 ...
```

```
## $ PROPDGMGEXP: Factor w/ 19 levels "", "-", "?", "+", ...: 17 17 17 17 17 17 17 17 17 17 17 ...
## $ CROPDMG : num 0 0 0 0 0 0 0 0 0 0 ...
## $ CROPDMGEXP: Factor w/ 9 levels "", "?", "0", "2", ...: 1 1 1 1 1 1 1 1 1 1 ... ##
$ WFO : Factor w/ 542 levels "", " CI", "%SD", ...: 1 1 1 1 1 1 1 1 1 1 ...
## $ STATEOFFIC: Factor w/ 250 levels "", "ALABAMA, Central", ...: 1 1 1 1 1 1 1 1 1 1 1 ...
## $ ZONENAMES : Factor w/ 25112 levels "", "
"| __truncated__, ...: 1 1 1 1 1 1 1 1 1 1 ...
## $ 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 : Factor w/ 436781 levels "", "\t", "\t\t", ...: 1 1 1 1 1 1 1 1 1 1 1 ...
... ## $ REFNUM : num 1 2 3 4 5 6 7 8 9 10 ...
```

```
head(stormData)
```

STAT...	BGN_DATE	BGN_T...	TIME_Z...	COU...	COUNTYN...	ST...	EVTYPE
<dbl>	<fctr>	<fctr>	<fctr>	<dbl>	<fctr>	<fctr>	<fctr>
1	1 4/18/1950 0:00:00	0130	CST	97	MOBILE	AL	TORNAD
2	1 4/18/1950 0:00:00	0145	CST	3	BALDWIN	AL	TORNAD
3	1 2/20/1951 0:00:00	1600	CST	57	FAYETTE	AL	TORNAD
4	1 6/8/1951 0:00:00	0900	CST	89	MADISON	AL	TORNAD
5	1 11/15/1951 0:00:00	1500	CST	43	CULLMAN	AL	TORNAD
6	1 11/15/1951 0:00:00	2000	CST	77	LAUDERDALE	AL	TORNAD

6 rows | 1-9 of 38 columns

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
Variable	Description
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

```

```
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$EVTYPE <- gsub('.*TORND AO.*', 'TORNADO',
stormDataTidy$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

```
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

(https://d396qusza40orc.cloudfront.net/repdata%2Fpeer2_doc%2Fpd01016005curr.pdf)” (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 CROPDGMG variable is encoded by the CROPDGMGEXP variable.

The documentation also specifies that the PROPDMGEXP and CROPDGMGEXP 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  ##  11585
1      5    210      1      1      4     18      3      3
##      B      H      K      M  ##      40      7 231427  11327

table(toupper(stormDataTidy$CROPDGMGEXP))
```

##					
##	?	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

```

# 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$PROP_COST <- with(stormDataTidy, as.numeric(PROPDMG) *
  apply(PROP_DMGE, getMultiplier))/10^9
stormDataTidy$CROP_COST <-
  with(stormDataTidy, as.numeric(CROPDMG) * apply(CROP_DMGE,
    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$FATALITIES
+ stormDataTidy$INJURIES),
                             by = list(EVENT_TYPE = stormDataTidy$EVTYPE),
                             FUN = sum,
                             na.rm = TRUE)
healthImpactData[order(healthImpactData$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$PROP_COST
+ stormDataTidy$CROP_COST),
                                 by =
list(EVENT_TYPE = stormDataTidy$EVTYPE),
                                 FUN = sum,
                                 na.rm = TRUE)

```

```
damageCostImpactData <- damageCostImpactData[order(damageCostImpactData$DAMAGE_I
MPACT, decreasing = TRUE),]
```

Results

Event Types Most Harmful to Population Health

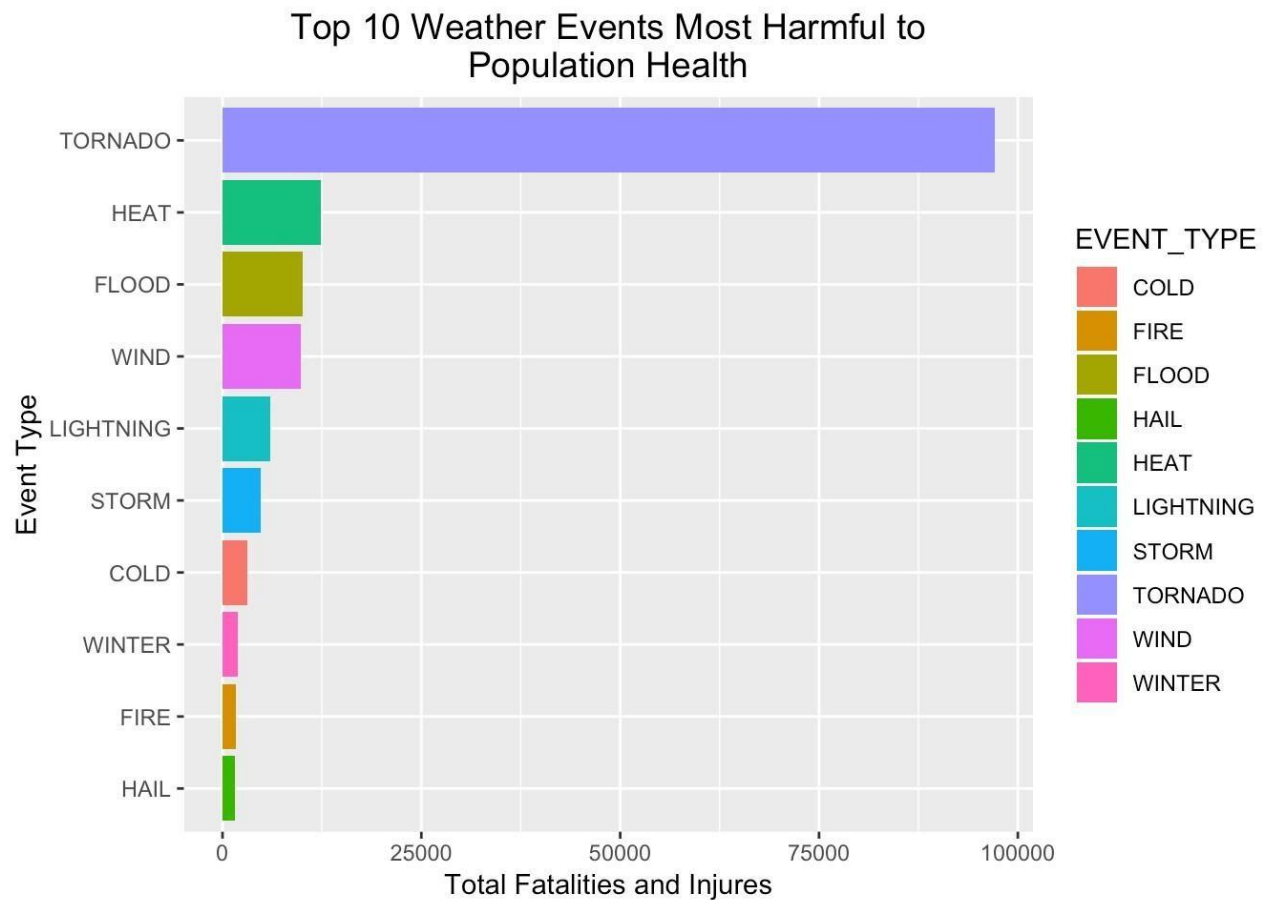
Fatalities and injuries have the most harmful impact on population health. The results below display the 10 most harmful weather events in terms of population health in the U.S.

```
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 = HEAL
TH_IMPACT, 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\nPopu lation 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.

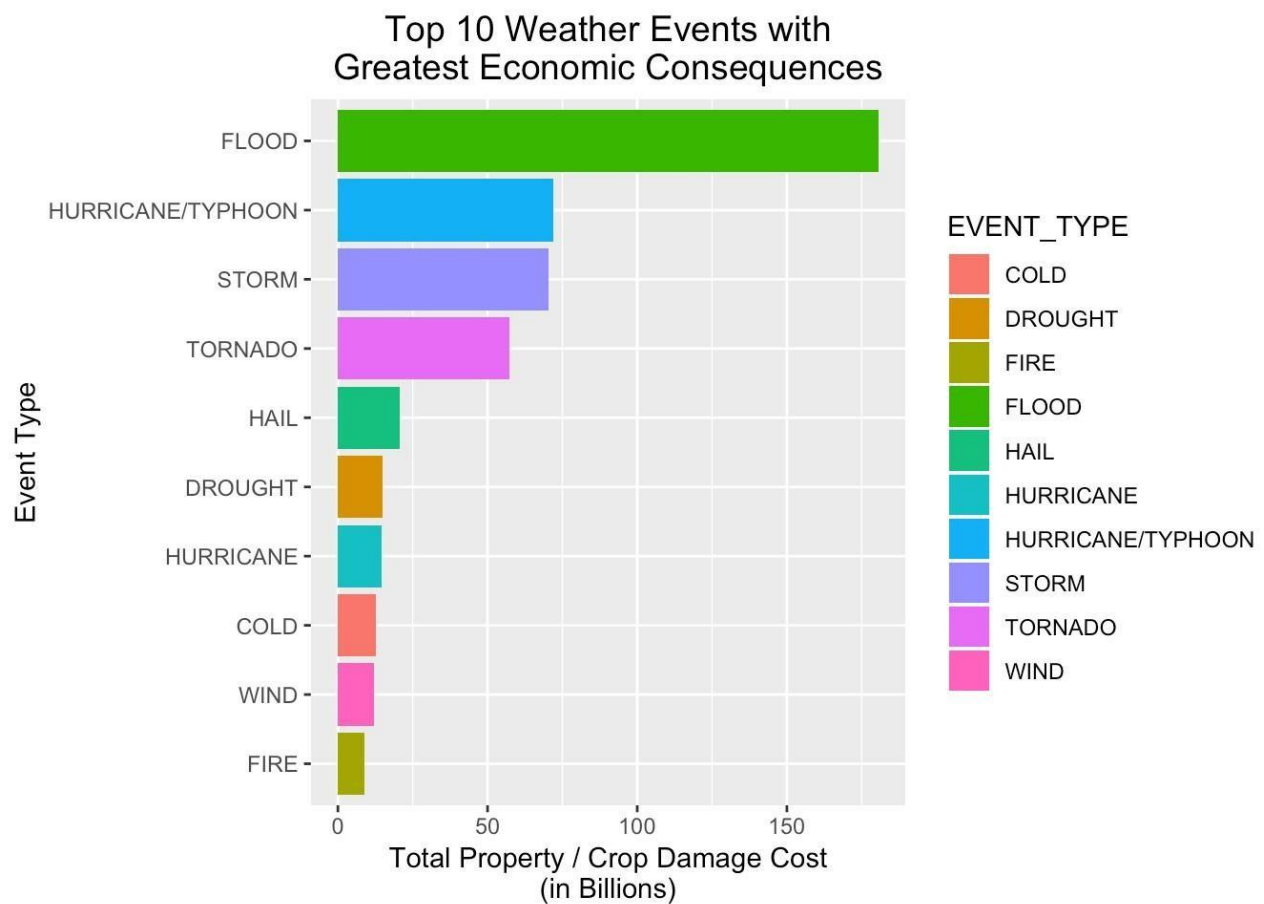
```
print(xtable(head(damageCostImpactData, 10),
                  caption = "Top 10
Weather Events with Greatest Economic Consequence s"),
      caption.placement = 'top',
      type = "html",
      include.rownames = FALSE,
      html.table.attributes='class="table-bordered", width="100%"')
```

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 = DAMA
GE_IMPACT, fill = EVENT_TYPE)) +
  coord_flip() +
  geom_bar(stat = "identity") +
  xlab("Event Type") +
  ylab("Total Property / Crop Damage Cost\n(in Billion s)") +
  theme(plot.title = element_text(size = 14, hjust =
0.5)) +
  ggtitle("Top 10 Weather Events with\nGreatest
Econom ic 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.