Effects of natural disasters to population health and economics in the United State (1950-2011)

Tanawut Noungneaw

2023-08-21

Synopsis

The processing and analysis data below are used to answer the following questions:

Q1. Across the United States, which types of events (as indicated in the EVTYPE variable) are most harmful with respect to population health?

Q2. Across the United States, which types of events have the greatest economic consequences?

This project involves exploring the U.S. National Oceanic and Atmospheric Administration's (NOAA) storm database.

Storm data.

This database tracks characteristics of major storms and weather events in the United States, including when and where they occur, as well as estimates of any fatalities, injuries, and property damage.

Loading necessary libraries

```r
library(dplyr)
library(ggplot2)
library(tidyr)
```

Loading and preprocessing the data

```r
# Set the url for download the data and path for its location
url <-"https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2"
path <- "stormdata.csv.bz2"
# Check the existent of data in zip file. If it isn't downloaded,
# this part will download it.
if(!file.exists("stormdata.csv.bz2")){
  download.file(url, path, method = "curl")
}
# Load  data
data <- read.csv("stormdata.csv.bz2")
```

Checking the datasets' dimensions and structure

```
# Dimensions checking
dim(data)
```

```
## [1] 902297     37
```

```
# Structure checking
str(data)
```

```
## '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 ...
```

Data processing

Subsetting data

```
# Select only the interested columns
cols = c("COUNTY", "STATE","EVTYPE","FATALITIES","INJURIES",
         "PROPDMG", "PROPDMGEXP", "CROPDMG", "CROPDMGEXP","BGN_DATE",
         "END_DATE")
usedata <- data[, cols]
# Filter no damage results off
subdata <- subset(usedata,  FATALITIES > 0 | INJURIES > 0 | PROPDMG > 0 |
                  CROPDMG >0)
```

Replacing multipliers with its values in scientific notation

The "PROPDMGEXP" and "CROPDMGEXP" columns represent the multiplier that multiply to the "PROPDMG" and "CROPDMG", respectively. According to the Storm Data Documentation page 12, the alphabet `B` stands for billions, `M` stands for millions, and `K` stands for thousands. In this case, there is an alphabet `H`, which I used for hundreds. These alphabets are used in the same way for lowercase and uppercase. The other symbol such as no spacing (`""`), `+`, `-`, and `?`, are replaced by `1` for multiplication.

```
# Property damage multiplier
table(toupper(subdata$PROPDMGEXP))
```

```
##
##             -      +      0      2      3      4      5      6      7      B
##  11585      1      5    210      1      1      4     18      3      3     40
##      H      K      M
##      7 231428  11327
```

```
# Crop damage multiplier
table(toupper(subdata$CROPDMGEXP))
```

```
##
##             ?      0      B      K      M
## 152664      6     17      7  99953   1986
```

```
# PROPDMGEXP
for (i in unique(subdata$PROPDMGEXP)){
  if(i %in% as.character(c(0:10))){
    subdata$PROPDMGEXP[subdata$PROPDMGEXP==i] <- paste("1e",i,sep = "")
  }
  else if(toupper(i) %in% c("K","H","M","B")){
    subdata$PROPDMGEXP[subdata$PROPDMGEXP=="h" |
                         subdata$PROPDMGEXP=="H"] <- paste("1e","2",sep = "")
    subdata$PROPDMGEXP[subdata$PROPDMGEXP=="k" |
                         subdata$PROPDMGEXP=="K"] <- paste("1e","3",sep = "")
    subdata$PROPDMGEXP[subdata$PROPDMGEXP=="m" |
                         subdata$PROPDMGEXP=="M"] <- paste("1e","6",sep = "")
    subdata$PROPDMGEXP[subdata$PROPDMGEXP=="b" |
```

```r
                         subdata$PROPDMGEXP=="B"] <- paste("1e","9",sep = "")
  }
  else {
    subdata$PROPDMGEXP[subdata$PROPDMGEXP==""|  subdata$PROPDMGEXP=="-" |
                         subdata$PROPDMGEXP=="+"] <- "1"
  }
}
# CROPDMGEXP
for (i in unique(subdata$CROPDMGEXP)){
  if(i %in% as.character(c(0:10))){
    subdata$CROPDMGEXP[subdata$CROPDMGEXP==i] <- paste("1e",i,sep = "")
  }
  else if(toupper(i) %in% c("K","M","B")){
    subdata$CROPDMGEXP[subdata$CROPDMGEXP=="k" |
                         subdata$CROPDMGEXP=="K"] <- paste("1e","3",sep = "")
    subdata$CROPDMGEXP[subdata$CROPDMGEXP=="m" |
                         subdata$CROPDMGEXP=="M"] <- paste("1e","6",sep = "")
    subdata$CROPDMGEXP[subdata$CROPDMGEXP=="b" |
                         subdata$CROPDMGEXP=="B"] <- paste("1e","9",sep = "")
  }
  else {
    subdata$CROPDMGEXP[subdata$CROPDMGEXP==""|  subdata$CROPDMGEXP=="?"] <- "1"
  }
}
# Change character class to numeric class
subdata$PROPDMGEXP <- as.numeric(subdata$PROPDMGEXP)
subdata$CROPDMGEXP <- as.numeric(subdata$CROPDMGEXP)

# Property damage multiplier
table(toupper(subdata$PROPDMGEXP))
```

```
##
##       1     100    1000   10000   1E+05   1E+06   1E+07   1E+09
##   11801       8  231429       4      18   11330       3      40
```

```r
# Crop damage multiplier
table(toupper(subdata$CROPDMGEXP))
```

```
##
##       1   1000  1E+06  1E+09
## 152687  99953   1986      7
```

Event names cleaning

The contents in column `EVTYPE` are really messed data. There are a lot of event names that can be categorized by the same group, but their spelling are incorrect. Some events can be categorized in many groups due to its overlapping. In this case, the Storm Data Documentation page 6 shows the 48 types of event that can be considered. The chuck below is used for cleaning this messed data by replacing it with the event names in the table from documentation. It may not be accurate due to the reasons that mentioned above. Lastly, I combined all of the events that cannot be grouped, and called, `Other`.

```
## Replacing event names
# Event table names
evnames <- 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 Hail", "Marine High Wind", "Marine Strong Wind",
             "Marine Thunderstorm Wind", "Rip Current", "Seiche", "Sleet",
             "Storm Surge/Tide", "Strong Wind", "Thunderstorm Wind", "Tornado",
             "Tropical Depression","Tropical Storm","Tsunami", "Volcanic Ash",
             "Waterspout", "Wildfire","Winter Storm","Winter Weather")
designator <- c("Z","Z","Z","Z","Z","C","Z","Z","Z","C","Z","Z","Z","C","C",
                "Z","C","Z","C","Z","C","Z","Z","Z","Z","Z","Z","Z","C","M",
                "M","M","M","Z","Z","Z","Z","Z","C","C","Z","Z","Z","Z","M",
                "Z","Z","Z")
event <- data.frame(EVTYPE = evnames, Designator = designator)
defined <- data.frame(Designator = c("C","Z","M"),
                      Def = c("Country/Parish", "Zone", "Marine"))
eventdf <- merge(x = event, y = defined, by = "Designator")
# Assign EVTYPE by the first event name
for (i in unique(eventdf$EVTYPE)) {
  subdata$EVTYPE[grep(paste("^", i ,sep = ""), subdata$EVTYPE,
                      ignore.case = TRUE)] <- i
}

# Wildfire
subdata$EVTYPE[grep("Fire", subdata$EVTYPE, ignore.case = TRUE)] <- "Wildfire"
# Thunderstorm wind
subdata$EVTYPE[grep("MARINE TSTM WIND", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Marine Thunderstorm Wind"
subdata$EVTYPE[grep("Non", subdata$EVTYPE, ignore.case = TRUE,
                    perl = TRUE)] <- "Strong Wind"
subdata$EVTYPE[grep("TSTM", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Thunderstorm Wind"
subdata$EVTYPE[grep("THUNDER", subdata$EVTYPE,
                    ignore.case = FALSE)] <- "Thunderstorm Wind"
# Coastal Flood, Flash Flood , Flood and Lakeshore Flood
subdata$EVTYPE[grep("Flash Flood", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Flash Flood"
subdata$EVTYPE[grep("(COASTAL).*FLOOD", subdata$EVTYPE,
                    ignore.case = TRUE,perl = TRUE)] <- "Coastal Flood"
subdata$EVTYPE[grep("(erosion).*FLOOD", subdata$EVTYPE,
                    ignore.case = TRUE,perl = TRUE)] <- "Coastal Flood"
subdata$EVTYPE[grep("(ICE JAM).FLOOD", subdata$EVTYPE,
                    ignore.case = TRUE,perl = TRUE)] <- "Flash Flood"
subdata$EVTYPE[grep("(SNOWMELT).FLOOD", subdata$EVTYPE,
                    ignore.case = TRUE,perl = TRUE)]  <- "Flash Flood"
subdata$EVTYPE[grep("(river|lake).*flood", subdata$EVTYPE,
                    ignore.case = TRUE,perl = TRUE)] <- "Lakeshore Flood"
subdata$EVTYPE[grep("^[^flood|coastal].*flood", subdata$EVTYPE,
```

```r
                    ignore.case = TRUE,perl = TRUE)] <- "Flash Flood"
subdata$EVTYPE[grep("tidal|stream", subdata$EVTYPE,
                    ignore.case = TRUE,perl = TRUE)] <- "Flash Flood"
# Surf
subdata$EVTYPE[grep("surf", subdata$EVTYPE,ignore.case = TRUE)] <- "High Surf"
# Heat
subdata$EVTYPE[grep("(excessive heat)", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Excessive Heat"
subdata$EVTYPE[grep("extreme heat", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Excessive Heat"
subdata$EVTYPE[grep("(?=.*heat)^(?!.*excessive)", subdata$EVTYPE,
                    ignore.case = TRUE,perl = TRUE)] <- "Heat"
# Tornado
subdata$EVTYPE[grep("TORN", subdata$EVTYPE, ignore.case = TRUE)] <- "Tornado"
# Thunder
subdata$EVTYPE[grep("THU", subdata$EVTYPE,
                    ignore.case = FALSE)] <- "Thunderstorm Wind"
subdata$EVTYPE[grep("TUN", subdata$EVTYPE,
                    ignore.case = FALSE)] <- "Thunderstorm Wind"
subdata$EVTYPE[grep("surge", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Storm Surge/Tide"
subdata$EVTYPE[grep("coastal.*(storm)", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Storm Surge/Tide"
# Winter
subdata$EVTYPE[grep("ICE STORM", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Ice Storm"
subdata$EVTYPE[grep("FREEZING RAIN", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Ice Storm"
subdata$EVTYPE[grep("FREEZING DRIZZLE", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Ice Storm"
subdata$EVTYPE[grep("EFFECT", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Lake-Effect Snow"
subdata$EVTYPE[grep("(?=.*ice)(?!.*storm)", subdata$EVTYPE,
                    ignore.case = TRUE, perl = TRUE)] <- "Winter Weather"
subdata$EVTYPE[grep("heavy .*snow", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Heavy Snow"
subdata$EVTYPE[grep("^(?!.*heavy)^(?!.*lake)(?=.*snow)", subdata$EVTYPE,
                    ignore.case = TRUE, perl = TRUE)] <- "Winter Weather"
subdata$EVTYPE[grep("(?=.*freez)(?!.*fog)(?!.*spray)", subdata$EVTYPE,
                    ignore.case = TRUE, perl=TRUE)] <- "Frost/Freeze"
subdata$EVTYPE[grep("frost", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Frost/Freeze"
subdata$EVTYPE[grep("Glaze", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Frost/Freeze"
# Cold
subdata$EVTYPE[grep("extreme cold", subdata$EVTYPE,
                    ignore.case = TRUE, perl=TRUE)] <- "Extreme Cold/Wind Chill"
subdata$EVTYPE[grep("(?=.*cold)^(?!.*extreme)", subdata$EVTYPE,
                    ignore.case = TRUE, perl=TRUE)] <- "Cold/Wind Chill"
subdata$EVTYPE[grep("(?=.*HAIL)^(?!.*marine)", subdata$EVTYPE,
                    ignore.case = TRUE, perl=TRUE)] <- "Hail"
subdata$EVTYPE[grep("icy", subdata$EVTYPE,
                    ignore.case = TRUE, perl=TRUE)] <- "Winter Weather"
```

```r
subdata$EVTYPE[grep("LOW TEMPERATURE",
                    subdata$EVTYPE, ignore.case = TRUE)] <- "Cold/Wind Chill"
subdata$EVTYPE[grep("BLIZZARD", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Blizzard"
subdata$EVTYPE[grep("COOL", subdata$EVTYPE,
                    ignore.case = TRUE, perl = TRUE)] <- "Cold/Wind Chill"
# Avalanche
subdata$EVTYPE[grep("AVAL", subdata$EVTYPE,
                    ignore.case = TRUE, perl=TRUE)] <- "Avalanche"
# Hurricane
subdata$EVTYPE[grep("Hurricane", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Hurricane (Typhoon)"
subdata$EVTYPE[grep("TYPHOON", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Hurricane (Typhoon)"
# Rain
subdata$EVTYPE[grep("RAIN", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Heavy Rain"
subdata$EVTYPE[grep("HEAVY SHOWER", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Heavy Rain"
# Lightning
subdata$EVTYPE[grep("lig", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Lightning"
# Wind
subdata$EVTYPE[grep("(?=.*HIGH)(?=.*WIND)^(?!.*marine)", subdata$EVTYPE,
                    ignore.case = TRUE,perl = TRUE)] <- "High Wind"
subdata$EVTYPE[grep("gust", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Strong Wind"
subdata$EVTYPE[grep("^WIND", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Strong Wind"
subdata$EVTYPE[grep("(?=.*wind)(?=.*chill)^(?=.*extreme)",subdata$EVTYPE,
                  ignore.case = TRUE,perl = TRUE)] <- "Extreme Cold/Wind Chill"
subdata$EVTYPE[grep("Whirlwind", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Dust Devil"
subdata$EVTYPE[grep("(?=.*MI)(?=.*wind)", subdata$EVTYPE,
                    ignore.case = TRUE,perl = TRUE)] <- "Strong Wind"
subdata$EVTYPE[grep("STORM FORCE WINDS", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Hurricane (Typhoon)"
subdata$EVTYPE[grep("Gradient", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "High Wind"
subdata$EVTYPE[grep("BLOWING DUST", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Dust Devil"
subdata$EVTYPE[grep("burst", subdata$EVTYPE,
                    ignore.case = TRUE)] <- "Strong Wind"

# High waves
subdata$EVTYPE[grep("(?=.*high)(?!.*wind)^(?!.*ASTRO)", subdata$EVTYPE,
                    ignore.case = TRUE, perl = TRUE)] <- "High Surf"
subdata$EVTYPE[grep("SEAS$", subdata$EVTYPE,
                    ignore.case = TRUE, perl = TRUE)] <- "High Surf"
subdata$EVTYPE[grep("wave", subdata$EVTYPE,
                    ignore.case = TRUE, perl = TRUE)] <- "High Surf"
subdata$EVTYPE[grep("SWELL", subdata$EVTYPE,
                    ignore.case = TRUE, perl = TRUE)] <- "High Surf"
```

```
# ETC (the remnants)
subdata$EVTYPE[grep("(?=.*FOG)^(?!.*freez)", subdata$EVTYPE,
                    ignore.case = TRUE, perl = TRUE)] <- "Dense Fog"
subdata$EVTYPE[grep("TURBULENCE", subdata$EVTYPE,
                    ignore.case = TRUE, perl = TRUE)] <- "High Wind"
subdata$EVTYPE[grep("Wintry", subdata$EVTYPE,
                    ignore.case = TRUE, perl = TRUE)] <- "Winter Weather"
subdata$EVTYPE[grep("mix", subdata$EVTYPE,
                    ignore.case = TRUE, perl = TRUE)] <- "Winter Weather"
subdata$EVTYPE[grep("PREC", subdata$EVTYPE,
                    ignore.case = TRUE, perl = TRUE)] <- "Heavy Rain"
subdata$EVTYPE[grep("Hypothermia", subdata$EVTYPE,
                    ignore.case = TRUE, perl = TRUE)] <- "Cold/Wind Chill"
subdata$EVTYPE[grep("Hyperthermia", subdata$EVTYPE,
                    ignore.case = TRUE, perl = TRUE)] <- "Heat"
subdata$EVTYPE[grep("LANDSPOUT", subdata$EVTYPE,
                    ignore.case = TRUE, perl = TRUE)] <- "Tornado"

# ?
subdata$EVTYPE[subdata$EVTYPE=="?"] <- "Other"
###############################################################################
# Matching all the event names, if it's not in events, then assign as "Other"
datasave <- subdata
for (i in unique(datasave$EVTYPE)) {
  subdata$EVTYPE[grep(paste("^", i ,sep = ""), subdata$EVTYPE,
                      ignore.case = TRUE)] <- i
}
for (i in unique(datasave$EVTYPE)) {
  if(!(i %in% eventdf$EVTYPE)){
    datasave$EVTYPE[grep(i, datasave$EVTYPE,
                         ignore.case = TRUE, perl = TRUE)] <- "Other"
  }
}
# Print all cleaned event types
table((datasave$EVTYPE))
```

```
##
##    Astronomical Low Tide                Avalanche                 Blizzard
##                        2                      269                      256
##            Coastal Flood          Cold/Wind Chill                Dense Fog
##                      236                      164                      181
##              Dense Smoke                  Drought               Dust Devil
##                        1                      276                      100
##               Dust Storm           Excessive Heat  Extreme Cold/Wind Chill
##                      104                      716                      330
##              Flash Flood                    Flood             Freezing Fog
##                    22275                    10528                        7
##             Frost/Freeze             Funnel Cloud                     Hail
##                      176                       13                    26164
##                     Heat               Heavy Rain               Heavy Snow
##                      255                     1158                     1397
##                High Surf                High Wind      Hurricane (Typhoon)
##                      247                     6208                      234
##                Ice Storm         Lake-Effect Snow          Lakeshore Flood
##                      797                      198                      133
##                Lightning              Marine Hail         Marine High Wind
##                    13303                        2                       19
##       Marine Strong Wind Marine Thunderstorm Wind                    Other
##                       46                      142                      276
##              Rip Current                   Seiche                    Sleet
##                      643                        9                        2
##         Storm Surge/Tide              Strong Wind        Thunderstorm Wind
##                      230                     3678                   119675
##                  Tornado      Tropical Depression           Tropical Storm
##                    39964                       35                      421
##                  Tsunami             Volcanic Ash               Waterspout
##                       14                        2                       63
##                 Wildfire             Winter Storm           Winter Weather
##                     1258                     1510                      916
```

```
# date time cleaning
# Change begin date and end date from character class to date class
datasave$BGN_DATE <- as.Date(datasave$BGN_DATE, format = "%m/%d/%Y %H:%M:%S")
datasave$END_DATE <- as.Date(datasave$END_DATE, format = "%m/%d/%Y %H:%M:%S")
# Add new column called REC_DATE to determine the period of recording
datasave$REC_DATE <- datasave$END_DATE - datasave$BGN_DATE
# Add new row to specify the event name "Other" as Undefined event
eventdf <- rbind(eventdf,c("U","Other","Undefinded"))
# Merge datasets with designators
alldata <- merge(x = datasave, y = eventdf, by = "EVTYPE")
## summaries health and economics damage
sumevents <- alldata %>%
            group_by(EVTYPE, Def) %>%
            summarise(SUMFATALITIES = sum(FATALITIES),
                      SUMINJURIES = sum(INJURIES),
                      SUMHEALTH = sum(SUMFATALITIES + SUMINJURIES),
                      SUMPROP = sum(PROPDMG*PROPDMGEXP),
                      SUMCROP = sum(CROPDMG*CROPDMGEXP),
                      SUMECODMG = sum(SUMPROP+SUMCROP))
```

```
## ‘summarise()‘ has grouped output by ’EVTYPE’. You can override using the
## ‘.groups‘ argument.
```

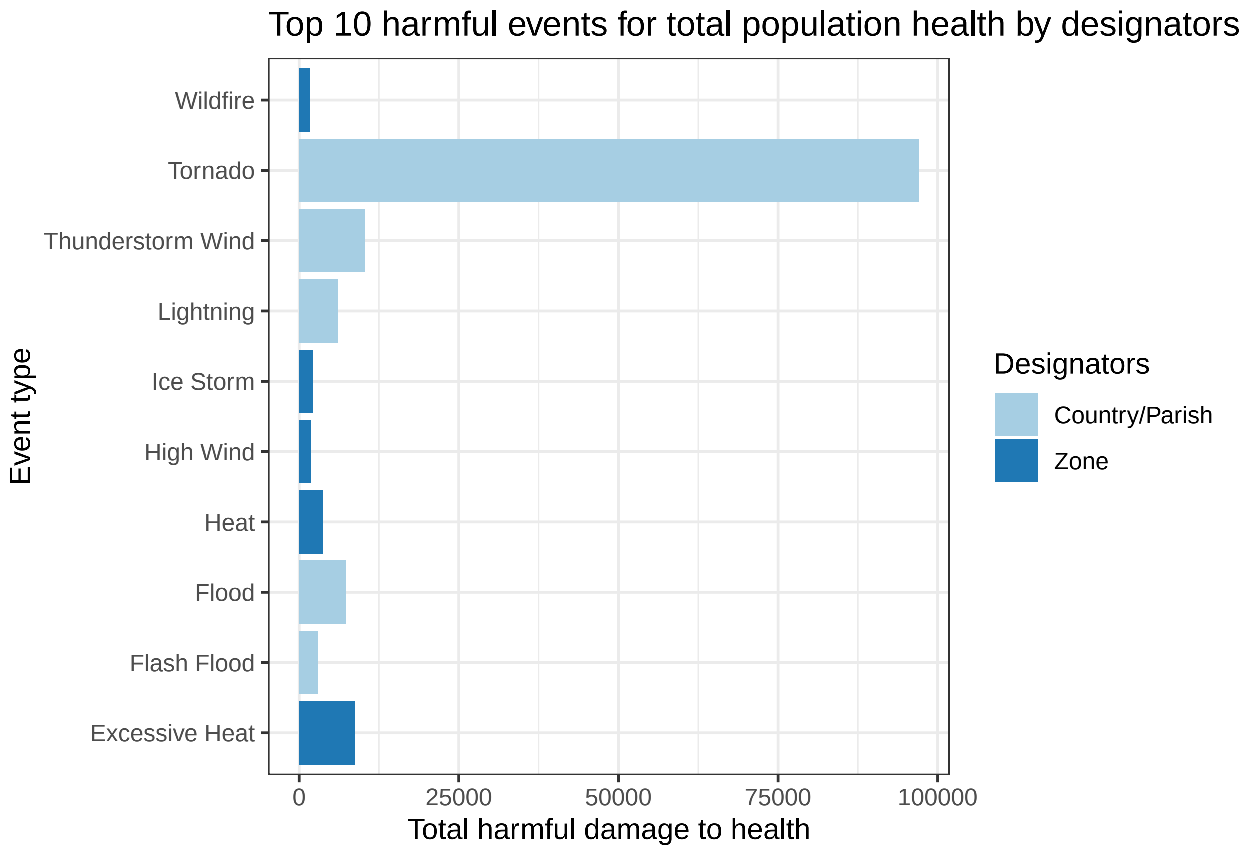
```
# Change column name to use in next part
colnames(sumevents)[2] <- "Designators"
```

Results

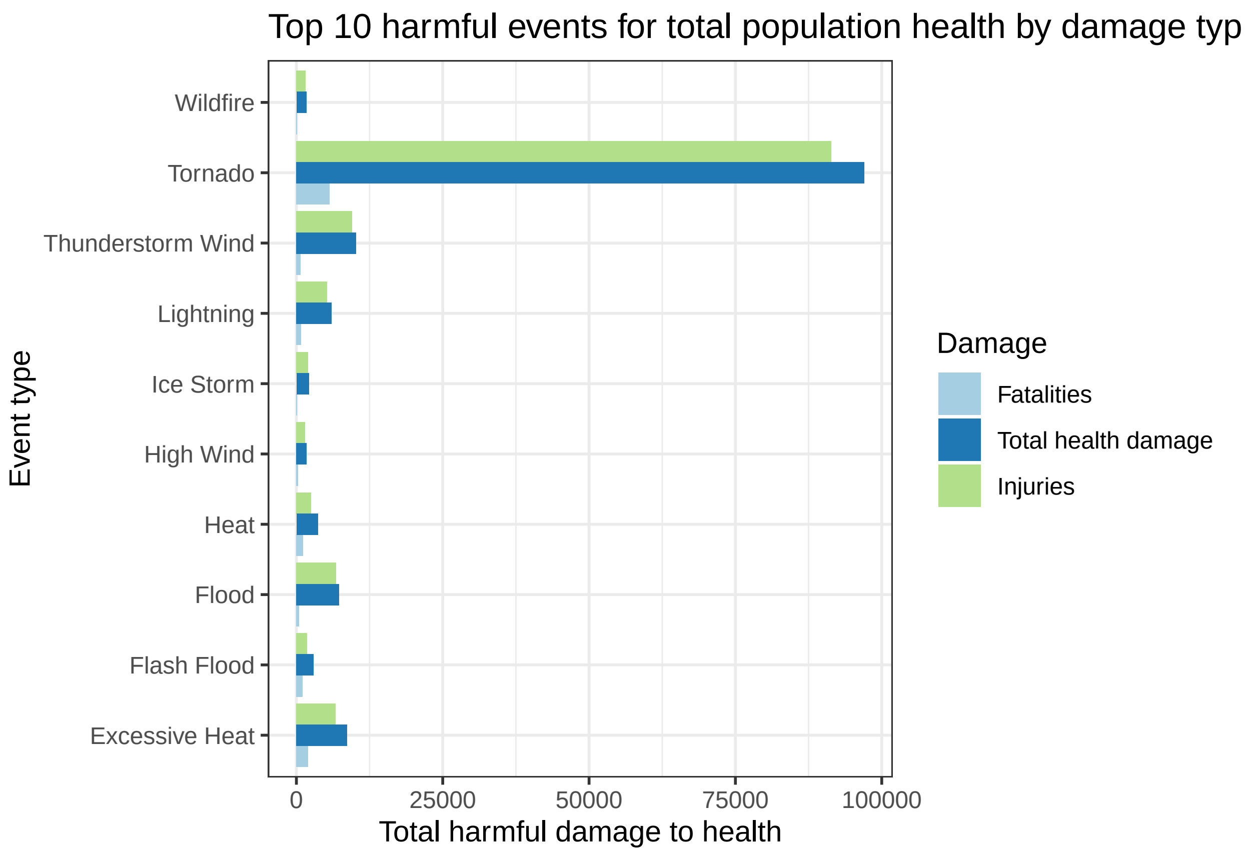
Total harmful damage to health

```
# Total health damage (top 10) by designators
ordsumevents <- sumevents %>% arrange(desc(SUMHEALTH))
ordsumevents <- ordsumevents[1:10,]
```

```r
g <- ggplot(ordsumevents, aes(x = SUMHEALTH, y = EVTYPE, fill = Designators))
g + geom_bar(stat = "identity") +
  xlab("Total harmful damage to health") +
  ylab("Event type") +
  ggtitle("Top 10 harmful events for total population health by designators") +
  scale_fill_brewer(palette = "Paired") +
  theme_bw()
```



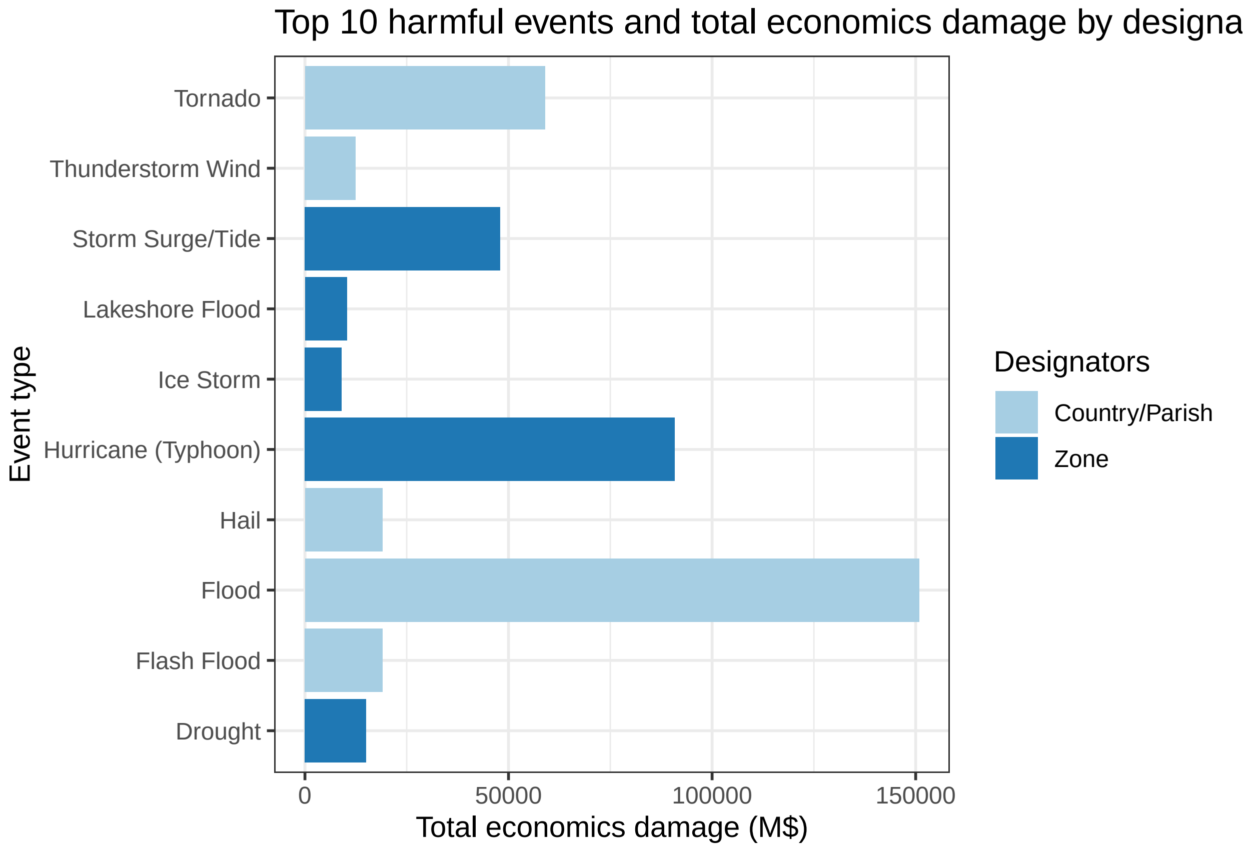
```r
# Total health damage (top 10) by each type of harmful damage
tose <- ordsumevents %>%
        pivot_longer(names_to = "Damage", values_to = "Value",
                     cols = c("SUMFATALITIES", "SUMINJURIES", "SUMHEALTH"))
g <- ggplot(tose, aes(x = Value, y = EVTYPE, fill = Damage))
g + geom_col(position = "dodge") +
  xlab("Total harmful damage to health") +
  ylab("Event type") +
  ggtitle("Top 10 harmful events for total population health by damage type") +
  scale_fill_brewer(palette = "Paired",
                    labels=c("Fatalities", "Total health damage", "Injuries")) +
  theme_bw()
```



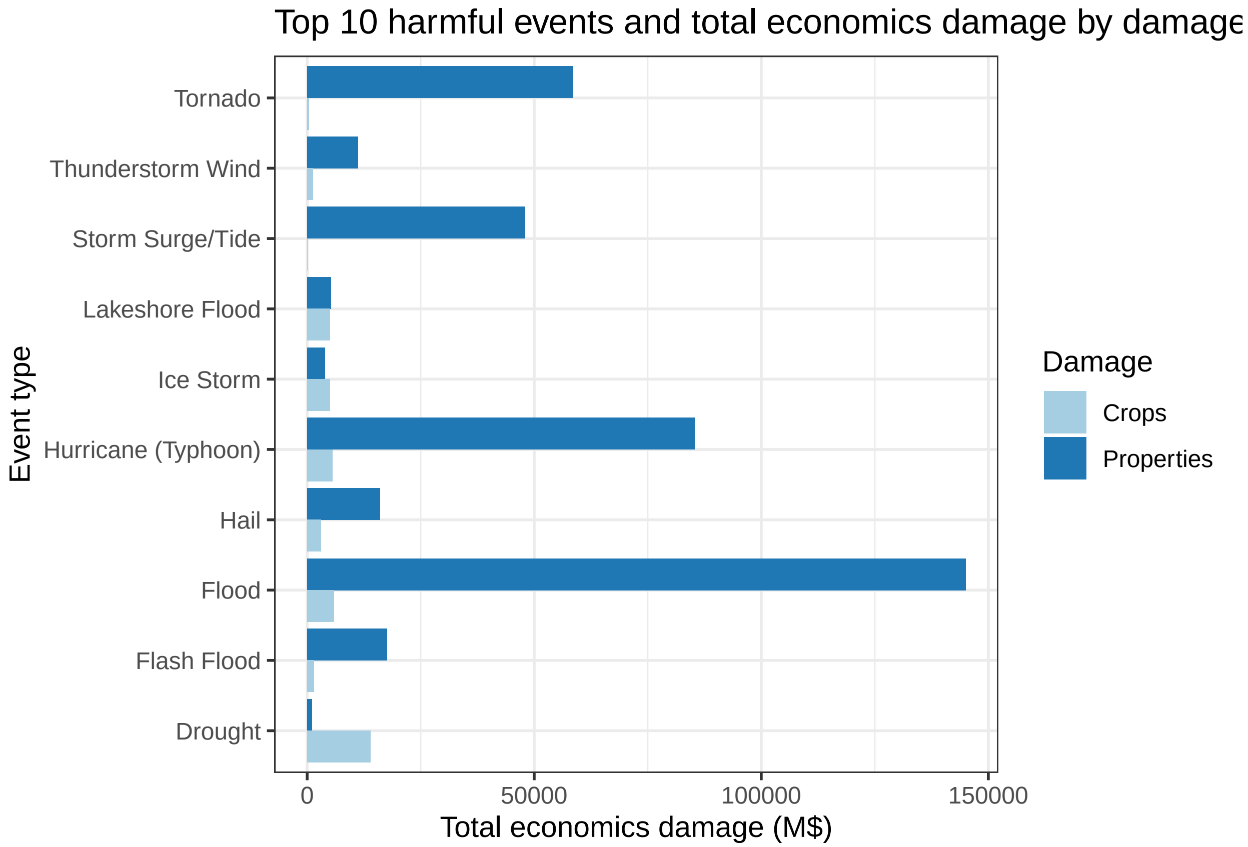
From the two figures above, tornado is the most harmful event for population health. It comes with the highest fatalities and injuries (almost 100,000) compare to the other events. The other events have the number of health damages lower than 12,5000.

Total economics damage

```r
# Total economics damage (top 10) by designator
ordsumevents <- sumevents %>% arrange(desc(SUMECODMG))
ordsumevents <- ordsumevents[1:10,]
g <- ggplot(ordsumevents,
            aes(x = SUMECODMG/1e6, y = EVTYPE, fill = Designators))
g + geom_bar(stat = "identity") +
  xlab("Total economics damage (M$)") +
  ylab("Event type") +
  ggtitle("Top 10 harmful events and total economics damage by designators") +
  scale_fill_brewer(palette = "Paired") +
  theme_bw()
```



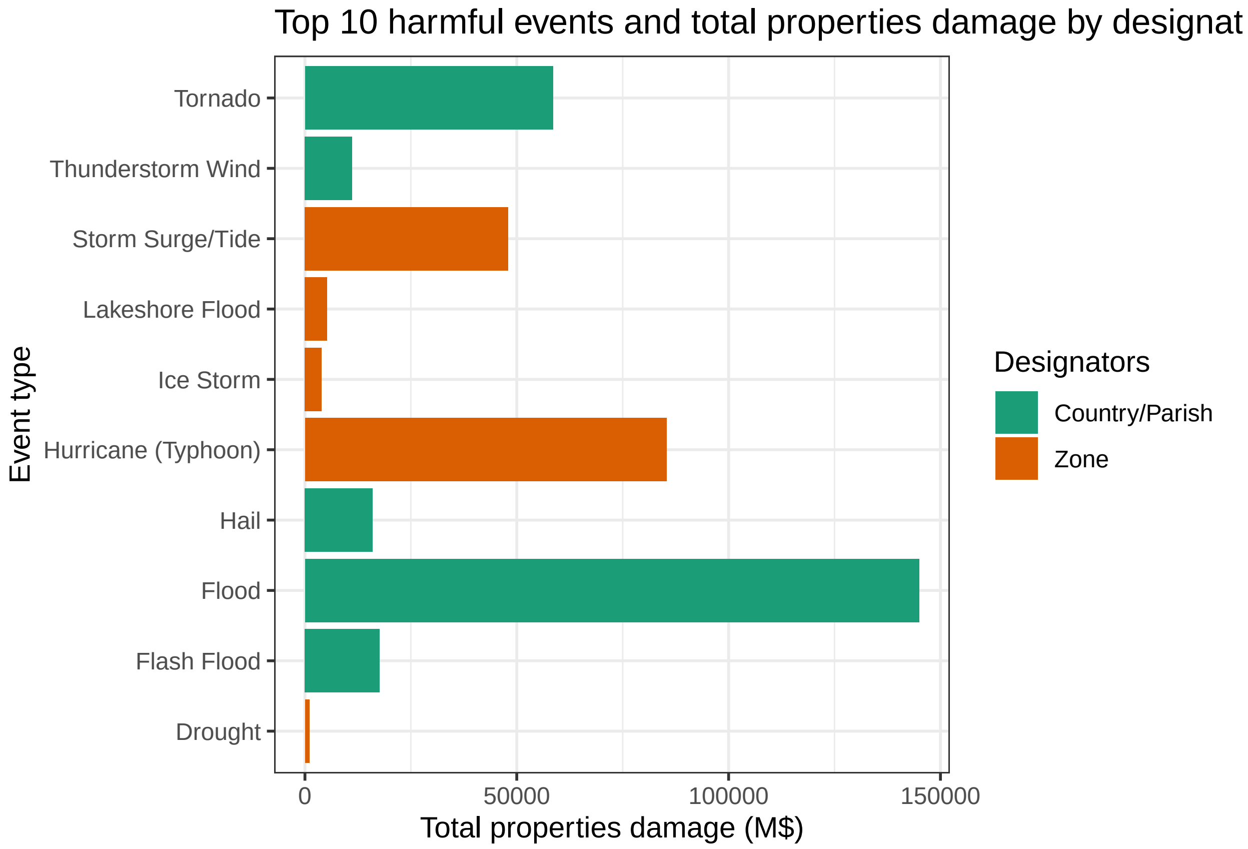
```
# Total economics damage (top 10) by each type of harmful damage
tose <- ordsumevents %>%
        pivot_longer(names_to = "Damage", values_to = "Value",
                     cols = c("SUMPROP", "SUMCROP"))
g <- ggplot(tose, aes(x = Value/1e6, y = EVTYPE, fill = Damage))
g + geom_col(position = "dodge") +
  xlab("Total economics damage (M$)") +
  ylab("Event type") +
  ggtitle("Top 10 harmful events and total economics damage by damage type") +
  scale_fill_brewer(palette = "Paired", labels=c("Crops", "Properties")) +
  theme_bw()
```



From the two figures above, it show that floods have the greatest consequences to the economics than the other damages, especially, the damage to properties. The second and third places are hurricane and tornado, respectively. Then, the properties and crops damage have been considered to be examined individually which show the results below.

Total properties damage

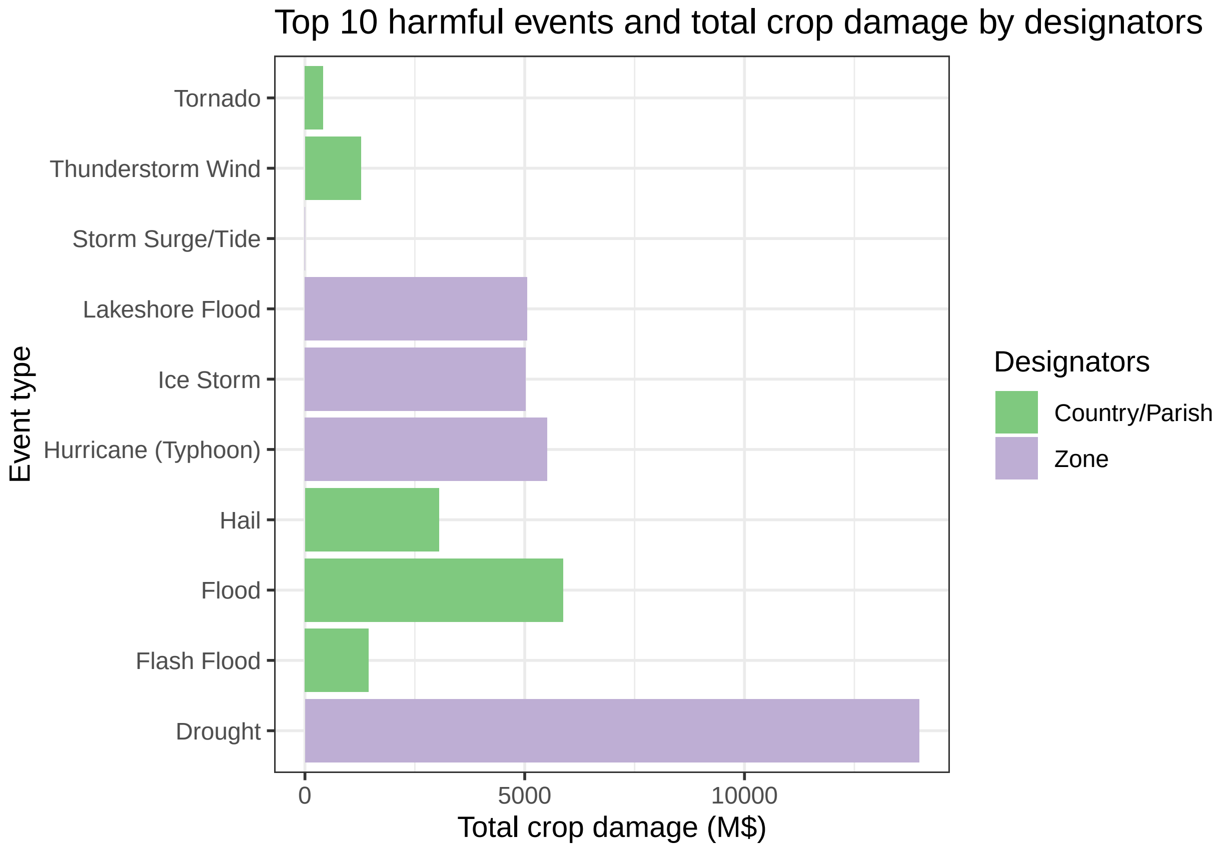
```
# Total properties damage (top 10)
ordsumevents <- ordsumevents %>% arrange(desc(SUMPROP))
ordsumevents <- ordsumevents[1:10,]
g <- ggplot(ordsumevents, aes(x = SUMPROP/1e6, y = EVTYPE, fill = Designators))
g + geom_bar(stat = "identity") +
  xlab("Total properties damage (M$)") +
  ylab("Event type") +
  ggtitle("Top 10 harmful events and total properties damage by designators") +
  scale_fill_brewer(palette = "Dark2") +
  theme_bw()
```



The top 10 total properties damage shows the same event type and trend as the economics consequences results, but different in total damages.

Total crops damage

```r
# Total crop damage (top 10)
ordsumevents <- ordsumevents %>% arrange(desc(SUMCROP))
ordsumevents <- ordsumevents[1:10,]
g <- ggplot(ordsumevents, aes(x = SUMCROP/1e6, y = EVTYPE, fill = Designators))
g + geom_bar(stat = "identity") +
  xlab("Total crop damage (M$)") +
  ylab("Event type") +
  ggtitle("Top 10 harmful events and total crop damage by designators") +
  scale_fill_brewer(palette = "Accent") +
  theme_bw()
```



The top 10 total crop damage is different from the economics and properties damage. In this case, the greatest damage is created by drought, while the flood got the second place.

Conclusion

To conclude, tornado was the greatest harmful event that affects United State population from year 1950 to November 2011. It caused almost 100,000 injuries and casualties in total. For economics, the greatest harmful event was flood, which caused around 150B, especially, for properties damage. However, the highest damaged for crop individually was caused by drought instead of flood, which caused around 15B.